Lecture #26: TA led Review

Singular Value Decomposition

$\underline{\underline{A}}_{mxn} = \underline{\underline{U}}_{mxm}\,\underline{\underline{S}}_{mxn}\,\underline{\underline{V}}_{nxn}^T$ $\qquad \underline{\underline{U}}^{-1} = \underline{\underline{U}}^T \qquad \underline{\underline{V}}^{-1} = \underline{\underline{V}}^T$

$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = \underbrace{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}_{U}\underbrace{\begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ 0 & & & 0 \end{bmatrix}}_{S}\underbrace{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}_{V^T}$$

$\underline{\underline{A}} = \underline{\underline{U}}\cdot\underline{\underline{S}}\cdot\underline{\underline{V}}^T$ $\qquad \sigma_1 \sim 0$

$\underline{\underline{U}}^T\cdot\underline{\underline{A}} = (\underline{\underline{U}}^T\underline{\underline{U}})\underline{\underline{S}}\cdot\underline{\underline{V}}^T$

$$\underline{\underline{S}}^{-1} = \begin{bmatrix} 1/\sigma_1 & & & 0 & 0 \\ & 1/\sigma_2 & & & \\ & & 1/\sigma_3 & & \\ 0 & & & \ddots & 0 \end{bmatrix}$$

$1/\sigma_1 \rightarrow \infty$ (treat it as zero)

$\underline{\underline{U}}^T\cdot\underline{\underline{A}} = \underline{\underline{I}}\cdot\underline{\underline{S}}\cdot\underline{\underline{V}}^T$

$\underline{\underline{V}}\cdot\underline{\underline{S}}^{-1}\underline{\underline{U}}^T\cdot\underline{\underline{A}} = \underline{\underline{I}}\cdot\underline{\underline{S}}\cdot\underline{\underline{V}}^T\underline{\underline{V}}\cdot\underline{\underline{S}}^{-1}$

$\underline{\underline{A}}^{-1} = \underline{\underline{V}}\cdot\underline{\underline{S}}^{-1}\underline{\underline{U}}^T$

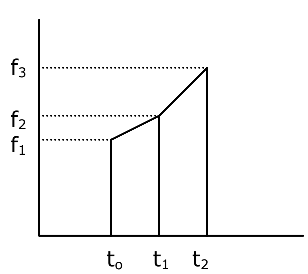
$\underline{\underline{A}}\cdot\underline{x} = \underline{b} \qquad \underline{x} = \underline{\underline{A}}^{-1}\underline{b}$

$\underline{x} = \underline{\underline{V}}\cdot\underline{\underline{S}}^{-1}\underline{\underline{U}}^T\cdot\underline{b}$

Ordinary Differential Equations

$d\underline{x}_{nm}/dt = \underline{F}_n(\underline{x}) \qquad \underline{x} = \underline{x}(t_o) + \int_{t_o}^{t} F(\underline{x}(t))dt$

Trapezoidal Rule:



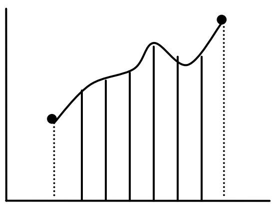
Error = $O(\Delta t^3)xN \rightarrow O(\Delta t^2)$

Figure 1. Integration by the Trapezoidal Rule.

Simpson's Error

Error = $O(\Delta t^5)xN \rightarrow O(\Delta t^4)$

Romberg Method (Richardson Extrapolation)



$$\underline{F}_{true} = \int_{t_0}^{t_n} f(t)dt \sim F(N) + C(\Delta t)^2 + \ldots$$

$$\underline{F}_{true} = \int_{t_0}^{t_n} f(t)dt \sim F(2N) + C\left(\frac{\Delta t}{2}\right)^2 + \ldots$$

$$\underline{F}_{true} = \frac{4}{3}F(2N) - \frac{1}{3}F(N) + O\left((\Delta t)^3\right)$$

Figure 2. Integration by the Romberg Method.

N → Δt

2N → $^{\Delta t}/_2$

ODE Solvers

Explicit Euler

$\underline{x}_{n+1} = \underline{x}_n + \underline{F}(\underline{x}_n)h + O(h^2)$

Runge Kutta

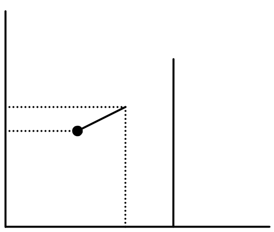


Figure 3. Runge Kutta Integration of differential equations.

Runge-Kutta-order 5 → 6 function evaluations per time steps

use intermediate value to calculate

Ode45 uses R-K 6 function evaluation

Stiff differential equation

$\underline{x} = a \cdot e^{-t} + b \cdot e^{-1000000t}$ { rate of time change are 1,000,000 times different

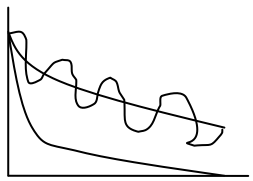


Figure 4. Example solution to differential equation.

must use many time steps; if big steps are used, you will oscillate around the solution.

$$\frac{d\underline{x}}{dt} = -\underline{\underline{c}} \cdot \underline{x}$$

$\Delta t < 2/\lambda_{max}$

Must use Implicit Method

Predictor-Corrector Method

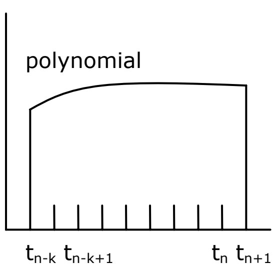


Figure 5. Predictor-corrector method.

DAE

$$\underline{\underline{M}}(\underline{X})\left(\frac{d\underline{x}}{dt}\right) = \underline{F}(\underline{X})$$

$$\begin{bmatrix} & & \\ & & \\ 0 & 0 & 0 \end{bmatrix} = f(\underline{x})$$ “ode23t,” “ode15i”

Optimization

$\min_{\underline{x}} f(\underline{x})$

$g(\underline{x}) = 0 \quad i = 1 \ldots n_e$

$h(\underline{x}) \geq 0 \quad i = 1 \ldots n_i$ CONSTRAINTS

If no constraints:

Gradient Method: $\underline{\nabla} f = \begin{pmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \partial f / \partial x_n \end{pmatrix}$ If no gradient given: $f_{minsearch}$

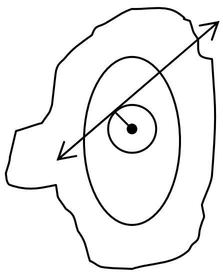


Figure 6. Gradient method contours.

Conjugate gradient method

As you get closer to the minimum, Newton's Method gives good convergence:

$\underline{\nabla} x_n = -H_n^{-1} \cdot \underline{\nabla} f_n$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & & & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

No Constraints: Broyden-Fletcher-Goldfarb-Shanno Method (BFGS)

With Constraints

$\underline{\nabla} f \rightarrow \underline{\nabla} g_i(\underline{x})$ Lagrangian

$\underline{\nabla} f = \sum_{i=1}^{n_e} \lambda_i \nabla g_i(\underline{x})$ $\underline{\underline{L}}(\underline{x}, \lambda) = f \sum_{i=1}^{n_e} \lambda_i \nabla g_i(\underline{x})$

$\nabla_{\underline{x}} \underline{\underline{L}} = 0$ $\nabla_{\underline{\lambda}} \underline{\underline{L}} = 0$

$\underline{\underline{L}}(\underline{x}, \underline{\lambda}) = f - \Sigma \lambda_i g_i(\underline{x}) + \Sigma (1/2\mu_i)[g_i(\underline{x})]^2$

KKT

$\underline{\underline{L}}(\underline{x},\underline{\lambda},\underline{k}) = f(\underline{x}) - \sum_{i=1}^{n_e} \lambda_i g_i(x) - \sum_{i=1}^{n_i} k_i h_i(x)$

$\nabla \underline{\underline{L}} = 0$ $\qquad g_i(\underline{x}) = 0$

$h_i(\underline{x}) \geq 0$ $\qquad k_i \geq 0 \quad h_j k_j = 0$

Sequential Quadratic Programming (SQP)

$\min_{\underline{x}} f(\underline{x})$

$c_m(\underline{x}) - s_m = 0$

for equality constraints: $s_m = 0$

for nonequality constraints: $s_m \geq 0$

Boundary Value Problems

$$v_x \frac{\partial \phi}{\partial x} + v_y \frac{\partial \phi}{\partial y} + v_z \frac{\partial \phi}{\partial z} = D\left[\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}\right] + S(\phi)$$

$$\left.\frac{\partial \phi}{\partial x}\right|_{x_n} = \frac{\phi_n - \phi_{n-1}}{x_n - x_{n-1}} + O(\Delta x) \qquad \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{x_n} = \frac{\phi_{n-1} - 2\phi_n + \phi_{n+1}}{2\Delta x}$$

Types of BC: $\varphi(\underline{x}_o) = \varphi_o$; $\left.\frac{\partial \phi}{\partial x}\right|_{x_n} = \frac{-3\phi_o + 4\phi_1 - \phi_2}{2\Delta x} = 0$